Locally Adaptive Greedy Approximations for Anisotropic Parameter Reduced Basis Spaces

Yvon Maday, Benjamin Stamm

LOCALLY ADAPTIVE GREEDY APPROXIMATIONS FOR ANISOTROPIC PARAMETER REDUCED BASIS SPACES

YVON MADAY* AND BENJAMIN STAMM†

Abstract. Reduced order models, in particular the reduced basis method, rely on empirically built and problem dependent basis functions that are constructed during an off-line stage. In the on-line stage, the precomputed problem dependent solution space can then be used in order to reduce the size of the computational problem. For complex problems, the number of basis functions required to guarantee a certain error tolerance can become too large in order to benefit computationally from the model reduction. To overcome this, the present work introduces a framework where local approximation spaces (in parameter space) are used to define the reduced order approximation in order to have explicit control over the on-line cost. This approach also adapts the local approximation spaces to local anisotropic behavior in the parameter space. We present the algorithm and present numerous numerical tests.

Key words. model reduction, reduced basis method, greedy algorithm

1. Introduction.

The recent progresses in the numerical simulation of physical phenomena obtained through the combination of robust and accurate approximation methods and faster and larger computational platforms have permitted to investigate more and more complex problems with improved reliability. These progresses have, in turn, lead to new demands for the numerical simulations that are not only used to understand a given state but investigate control and optimization problems. Here it is not only a unique (or very few) instance(s) of a given model that is required but a large number of similar computations of a now parameter-dependent model dynamically varying through the values of the parameter under which the behavior of the solution is sought for.

Faster solution algorithms are often not sufficient to achieve these new demands in many engineering applications and reduced numerical methods have been proposed as surrogates to standard numerical approximations of given mathematical models. Among these methods, the proper orthogonal decomposition (POD) and reduced basis (RB) methods are two main classes of higher-order mathematical techniques that have been widely developed during the last decade. These approaches have gained in generality and reliability [2, 3, 11, 12].

The basic idea behind these reduced numerical methods is the notion of small Kolmogorov n-width of the set of all solutions obtained when the parameters varies. Indeed POD or RB methods are based on a two steps strategy : the first step (off-line stage) allows to select particular instances of the parameters, for which a very accurate approximation of the solution is computed : the associated solutions constitute the reduced basis, in a second step (the on-line stage) the generic solution (for other instances of the parameter) are approximated by a linear combination of these reduced basis elements. The interest of the approach lies in the fact that, in cases the Kolmogorov n-width is small — and there are many reasons a priori, such as regularity of the solutions as a function of the parameters, and a posteriori, revealed by a large number of simulation — these reduced model approximations require very few degrees of freedom and in some sense are close in spirit to other high order approximation

* UPMC Univ Paris 06, UMR 7598, Laboratoire Jacques-Louis Lions, F-75005, Paris, France, and Division of Applied Maths, Brown University, Providence RI, USA. (maday@ann.jussieu.fr)

†Department of Mathematics, University of California, Berkeley, CA 94720, USA (stamm@math.berkeley.edu).

methods like spectral methods, but with an improved efficiency.

It is well known that high order methods generally take advantage of a global approach by using basis functions that have a large support that, combined with important regularity of the solution to be approximated (going some times up to analyticity) allows to get, with very few degrees of freedom, a very good accuracy. For most practical design problems in engineering though, the solution is not analytical and most of the time regularity exists but only locally which precludes the interest of global approximations. This is the reason why, for instance, by breaking the global framework to locally piecewise global approaches, the spectral element methods reveals superiority with respect to plain spectral method: a trade off between locality and globality is generally preferred as is demonstrated in e.g. [4] for approximation in spacial direction by spectral element approximations. The same cause imply the same effects in the parameter directions for RB approximations, this the reason why recently, some works have been devoted to proposing ways of adding locality to the reduced methods yielding parameter subdomain domain refinement.

In this paper we shall focus on the certified RB framework for which, the construction of basis functions during the first stage of the algorithm is, as in most of the current approaches, performed through a greedy strategy based on a posteriori error estimator. This concerns both the Galerkin approximation and the empirical interpolation methods. For either high dimensional parameter spaces or spaces with large measure one may encounter the problem that the size of the reduced basis turns out to be larger than desired. Following the lines drawn above, a first idea in this context has been presented by Eftang, Patera and Rønquist [6] and also Eftang and Stamm [7] where the parameter space is decomposed into cells where different reduced basis sets are assembled. This approach presents clear advantages in the size of the matricial system that appears in the on-line solution procedure, and corroborates the natural feeling that, in order to approximate the solution at a given parameter, primarily those solutions in the reduced basis corresponding to parameters that are close to the parameter we are interested in are to be involved in the linear approximation.

A drawback of the current approach [6, 7] however is that, in two adjacent parameter-subdomains, some of the parameters that are selected may be very close. Due to the difference between the CPU time associated with exploiting a RB method in the on-line stage, and that required for constructing the basis elements in the off-line step, this leads to the idea that it might be interesting to be able to use, in one parameter subdomain, the parameters that are used in the adjacent ones. Parameter domain decomposition may not thus be the ultimate approach. Another drawback of all current greedy approach is to be unable to master the size of the discrete system that will be solved in the on-line procedure. These remarks have motivated us to investigate the alternative discussed in this paper.

An outline of the paper is as follows. In Section 2 we present the standard greedy algorithm that is widely used in reduced basis computations and introduce then in Section 3 the concept of local approximation spaces that are able to take into account the local anisotropies in the parameter space. The construction of the local approximation spaces is based on a semi-distance function which is constructed in Section 4 through a learning process of the geometry of the manifold of the solution set. Section 5 introduces the concept of a varying train set that is chosen in a optimal fashion using the possibly anisotropic semi-distance function. Finally, in Section 6 we explain some practical aspects of the on-line implementation and Section 7 presents some numerical examples.

2. The greedy algorithm for reduced basis approximations. Let us first introduce a classical greedy-algorithm to have a common ground to present our ideas. In a general setting, it is assumed that for each parameter value $\boldsymbol{\mu}$ in a parameter domain $\mathbb{P} \subset \mathbb{R}^p$, a $\boldsymbol{\mu}$-dependent function $v(\boldsymbol{\mu}) \in \mathbb{W}$ of the variable $\boldsymbol{x}$ in a bounded domain $\Omega \subset \mathbb{R}^d$ (e.g. $d = 2$ or $d = 3$) can be computed. The space $\mathbb{W}$ denotes some functional space of functions defined on Ω. In order to fix the ideas this can be either that $v(\boldsymbol{\mu})$ is a solution to a parameter dependent partial differential equation, an approximation of which can be computed by e.g. classical finite element or spectral method. It can also be that $v(\boldsymbol{\mu})$ is a function $f(u(\boldsymbol{\mu}))$ of some $u(\boldsymbol{\mu})$, the evaluation of $u(\boldsymbol{\mu})$ is easy at some few points in Ω but the general knowledge is required for further purposes. Let

$$\mathbb{S}_N = \{\boldsymbol{\mu}^1, \cdots, \boldsymbol{\mu}^N\}$$

be a collection of N parameters in $\mathbb{P}$ and

$$\mathbb{W}_N = \text{span}\{v(\boldsymbol{\mu}^1), \cdots, v(\boldsymbol{\mu}^N)\}$$

be the approximation space associated to the set $\mathbb{S}_N$. A projection/interpolation operator $P_N : \mathbb{W} \to \mathbb{W}_N$ is supposed to exists that can be either a parameter dependent Ritz-projection (reduced basis methods in case of a parameter dependent PDE) [12, 13], an interpolation operator (empirical interpolation methods in case $v(\boldsymbol{\mu})$ is a function of some $u(\boldsymbol{\mu})$, e.g. $v(\boldsymbol{\mu}) = [u(\boldsymbol{\mu})]^4$ or $v(\boldsymbol{\mu}) = e^{[u(\boldsymbol{\mu})]}$) [1, 8] or simply a L^2-projection. Further, it is assumed that for each parameter value $\boldsymbol{\mu} \in \mathbb{P}$ an error estimator $\eta(\boldsymbol{\mu}; \mathbb{W}_N)$ of the approximation of $v(\boldsymbol{\mu})$ by $P_N(v(\boldsymbol{\mu}))$ can be computed (this can be e.g. through a posteriori analysis of the residual, see eg [12]). The case where $\eta(\boldsymbol{\mu}; \mathbb{W}_N)$ describes the exact error is not excluded in this general framework.

The following sketch represent a typical greedy algorithm: Let $\Xi_{\texttt{trial}}$ be a chosen finite trial set $\Xi_{\texttt{trial}} \subset \mathbb{P}$ of representative points.

`ClassicalGreedy`

1. Choose (possibly randomly) $\boldsymbol{\mu}^1 \in \Xi_{\texttt{trial}}$, set $\mathbb{S}_1 = \{\boldsymbol{\mu}^1\}$, $\mathbb{W}_1 = \text{span}\{v(\boldsymbol{\mu}^1)\}$ and $N = 1$, $\texttt{err} = \max_{\boldsymbol{\mu} \in \Xi_{\texttt{trial}}} \eta(\boldsymbol{\mu}; W_1)$.
2. `While` $\texttt{err} > \texttt{tol}$
3. Find $\boldsymbol{\mu}^{N+1} = \text{argmax}_{\boldsymbol{\mu} \in \Xi_{\texttt{trial}}} \eta(\boldsymbol{\mu}; W_N)$, $\texttt{err} = \max_{\boldsymbol{\mu} \in \Xi_{\texttt{trial}}} \eta(\boldsymbol{\mu}; W_N)$.
4. Compute $v(\boldsymbol{\mu}^{N+1})$, set $\mathbb{S}_{N+1} = \mathbb{S}_N \cup \{\boldsymbol{\mu}^{N+1}\}$ and $\mathbb{W}_{N+1} = \text{span}\{\mathbb{W}_N, v(\boldsymbol{\mu}^{N+1})\}$.
5. Set $N := N + 1$.
6. `End while`.

Algorithm: Classical greedy algorithm.

REMARK 2.1. *The introduction of the finite set $\Xi_{\texttt{trial}}$ is due to practical implementation. Actually the definition of this finite set can evolve during the algorithm. We shall elaborate on this in the frame of our approach latter in the paper in Section 5.*

Let $\tilde{N}$ denote the final size of the reduced basis space $\mathbb{W}_{\tilde{N}}$ such that the accuracy criterion

$$\max_{\boldsymbol{\mu} \in \Xi_{\texttt{trial}}} \eta(\boldsymbol{\mu}; W_{\tilde{N}}) \leq tol$$

is satisfied. Note that this final size is a consequence of the tolerance that has been chosen, hence, for a prescribed tolerance, we have not the hand on the complexity of the RB approximation method.

3. Local approximations. The new approach pursued in this work is in the same spirit of partitioning the parameter set as in [6, 7], however (i) we do not impose a clear partition of the parameter space but rather collect a global set of sample points $\mathbb{S}$ — preliminary constructed (in an off-line stage) with a given tolerance requirement — (ii) we choose a priori the size of the system we want to solve online by selecting an integer N, then (iii) when a reduced basis approximation is to be computed for a certain given parameter value $\boldsymbol{\mu} \in \mathbb{P}$, we only use the N basis functions whose parameter values lie in a ball

$$B_{\boldsymbol{\mu}} = \{\tilde{\boldsymbol{\mu}} \in \mathbb{P} \,|\, d(\boldsymbol{\mu}, \tilde{\boldsymbol{\mu}}) \leq r(\boldsymbol{\mu})\}, \tag{3.1}$$

for a given semi-distance function $d(\cdot, \cdot)$. Thus, the radius $r(\boldsymbol{\mu})$ is computed in such a way that there are actually N basis functions in the ball. Therefore the local sample space is defined by

$$\mathbb{S}_{\boldsymbol{\mu}} = B_{\boldsymbol{\mu}} \cap \mathbb{S} = \{\tilde{\boldsymbol{\mu}} \in \mathbb{S} \,|\, \tilde{\boldsymbol{\mu}} \in B_{\boldsymbol{\mu}}\}$$

with cardinality equal to N[1]. Further the local reduced basis approximation space shall be defined by $\mathbb{W}_{\boldsymbol{\mu}} = \operatorname{span}\{v(\tilde{\boldsymbol{\mu}}) \,|\, \tilde{\boldsymbol{\mu}} \in \mathbb{S}_{\boldsymbol{\mu}}\}$ and its associated local projection by $P_{\boldsymbol{\mu}} : \mathbb{W} \to \mathbb{W}_{\boldsymbol{\mu}}$.

One basic ingredient of this local reduced basis approximation is thus the metric $d(\cdot, \cdot)$; the radius is adjusted to select the total number of modes N for each value of $\boldsymbol{\mu}$ whereas the metric can be chosen isotropic or better accounts for anisotropies in the parameter space. The construction of an adapted local metric is a topic itself and is addressed in the upcoming Section 4.

The basic principle of the Locally Adaptive Greedy Approximation is to first select the number N of basis functions we want to get involved in the on-line procedure (i.e. the size of the matrix system to be solved on-line). A standard greedy algorithm, as described by Section 2, is then performed until $N + 1$ basis functions are selected. Then, the second stage of the algorithm is performed. The basis functions are no longer searched globally but in the above described local balls. The construction of those requires the data of a metric the construction of which is explained in Section 4 (note that, by default, a uniform metric is possible). This metric is updated at each one of the steps of this second stage. Having such a current metric, only local basis functions are considered in order to enrich the set $\mathbb{S}_N$: the locality of the basis functions affects the greedy selection of the basis functions since several basis functions can be chosen per iteration. The following procedure is proposed. At any given iteration, the first sample point, say $\overline{\boldsymbol{\mu}}$, is chosen such that the estimated error $\eta(\boldsymbol{\mu}; W_{\boldsymbol{\mu}})$ is maximized over the parameter domain as in the traditional setting except that only a local approximation space $W_{\boldsymbol{\mu}}$ is considered to compute the error (-estimator). Next, the maximum estimated error $\eta(\boldsymbol{\mu}; W_{\boldsymbol{\mu}})$ is searched over all parameter values lying outside of the domain of influence of $B_{\overline{\boldsymbol{\mu}}}$. By the domain of influence of $B_{\overline{\boldsymbol{\mu}}}$ we mean any $\boldsymbol{\mu}$ such that $\overline{\boldsymbol{\mu}} \in B_{\boldsymbol{\mu}}$ or $\boldsymbol{\mu} \in B_{\overline{\boldsymbol{\mu}}}$. Outside of the domain of influence of $B_{\overline{\boldsymbol{\mu}}}$ more basis functions can however be added. This condition assures that at most one sample

[1]Note that N has not the same meaning as in the previous section any more; hence the cardinal of $\mathbb{S}$ is not N but a $K > N$ generally much larger.

point is added per ball. This procedure is repeated, and more and more parts of the parameter domain are excluded, until the remaining set is the empty set. Given K global sample points $\mathbb{S}$ the following pseudo-code results for one iteration, again for a given subset $\Xi_{\texttt{trial}} \subset \mathbb{P}$ of representative points :

```
EnrichmentLoop
    1. Set Ξ̃_trial = Ξ_trial, err = max_{μ∈Ξ̃_trial} η(μ; W_μ).
    2. While Ξ̃_trial ≠ ∅ and err > tol
    3.     Set μ^{K+1} = argmax_{μ∈Ξ̃_trial} η(μ; W_μ), err = max_{μ∈Ξ̃_trial} η(μ; W_μ).
    4.     Compute v(μ^{K+1}) and set S_{K+1} = S ∪ {μ^{K+1}}.
    5.     Set Ξ̃_trial := Ξ̃_trial \ {μ ∈ Ξ_trial | μ ∈ B_{μ^{K+1}} or μ_{K+1} ∈ B_μ}.
    6.     Set K := K + 1.
    7. End while.
```

Algorithm: Basis enrichment loop.

4. Anisotropy and local metric - a general finite difference based approach. In order to reduce at most the number of elements involved in the reduced basis approximation, one has to estimate, on the fly, the way the (unknown) solution $v(\boldsymbol{\mu})$ depends on $\boldsymbol{\mu}$.

Indeed, for instance assuming that the parameter domain $\mathbb{P}$ is a subset of $\mathbb{R}^2$, so that $\boldsymbol{\mu} = (\mu_1, \mu_2)$, and assuming that there exists a function φ of one variable, such that $v(\boldsymbol{\mu}) = \varphi(\mu_1 + \mu_2)$, then the optimal selection of parameters should be sought e.g. along the line $\mu_1 = \mu_2$ involving only a one dimensional parameter set. Of course the fact that v depends only on $\mu_1 + \mu_2$ is generally not known. What we have to figure out from prior computations, during the greedy algorithm, is if in the solution dependency, there is such an anisotropy. This is a quite classical quest in numerical analysis, and the knowledge of this valuable information implies a large computational reduction. As in other instances, e.g. adaptive approaches like the one used in finite element approximations (see e.g. [5]), the goal is the construction of a (semi-) metric $d(\cdot, \cdot)$ that accounts for the anisotropic behavior of changes of $v(\boldsymbol{\mu})$ with respect to variations in $\boldsymbol{\mu}$. Similarly as in other approaches, the metric is derived from the knowledge of the Hessian of the function v with respect to the parameter $\boldsymbol{\mu}$.

We propose here a general framework to obtain an approximation of the Hessian by finite differences. We always assume that $\Xi_{\texttt{trial}}$ is such that its convex hull spans the parameter space $\mathbb{P}$.

[1] **Definition of the Hessian.** The goal is to define a Hessian matrix $\overline{H}(\boldsymbol{\mu})$ for each point $\boldsymbol{\mu} \in \Xi_{\texttt{trial}}$ upon which the metric will be based on. Since the Hessian is based on the reduced basis approximation, it is updated/constructed at each iteration in the algorithm. To do so, compute a reduced basis approximation $v_{\texttt{rb}}(\boldsymbol{\mu})$ of $v(\boldsymbol{\mu})$ based on the current reduced basis, on the stencil $\boldsymbol{\mu}(\alpha) = \boldsymbol{\mu} + \sum_{i=1}^{p} \alpha^i \delta\mu^i$, with $\boldsymbol{\mu} \in \Xi_{\texttt{trial}}$ and where $\delta\mu^i$, $i = 1, .., p$, is a positive small increment in the i^{th} direction in the parameter and $\alpha^i = -1, 0, 1$, such that, at most, two of them are non zero. From these 3^p approximations $v_{\texttt{rb}}(\boldsymbol{\mu}(\alpha))$ of $v(\boldsymbol{\mu}(\alpha))$ on the surrounding stencil $\boldsymbol{\mu}(\alpha)$ of $\boldsymbol{\mu}$, one can, by finite differences, evaluate an approximation of the Hessian of $v(\boldsymbol{\mu})$ as a function of $\boldsymbol{\mu}$ by taking e.g. $H_{ii}(\cdot; \boldsymbol{\mu}) = D_{ij} v_{\texttt{rb}}(\boldsymbol{\mu})$ on Ω where the operator D_{ij} is

defined by

$$D_{ij}v_{\mathtt{rb}}(\boldsymbol{\mu}) = \frac{v_{\mathtt{rb}}(\boldsymbol{\mu}+\delta\mu^i+\delta\mu^j) - v_{\mathtt{rb}}(\boldsymbol{\mu}-\delta\mu^i+\delta\mu^j) - v_{\mathtt{rb}}(\boldsymbol{\mu}+\delta\mu^i-\delta\mu^j) + v_{\mathtt{rb}}(\boldsymbol{\mu}-\delta\mu^i-\delta\mu^j)}{4\delta\mu^i\delta\mu^j}.$$

Remembering that $v_{\mathtt{rb}}(\boldsymbol{\mu})$ is a function on Ω, the above formula thus defines a Hessian matrix for each point $\boldsymbol{x} \in \Omega$ and parameter value $\boldsymbol{\mu} \in \mathbb{P}$. In order to obtain one Hessian $\overline{H}_{ij}(\boldsymbol{\mu})$ per parameter point $\boldsymbol{\mu} \in \mathbb{P}$ one needs to define an average operator over the space Ω. If $\{\varphi_n\}_{n=1}^N$ denotes the basis of $\mathbb{W}_{\boldsymbol{\mu}}$ used to build $v_{\mathtt{rb}}$ such that

$$v_{\mathtt{rb}}(\boldsymbol{x};\boldsymbol{\mu}) = \sum_{n=1}^{N} \alpha_n(\boldsymbol{\mu})\varphi_n(\boldsymbol{x}),$$

we propose to define an averaged Hessian over the Ω-space as follows

$$\underline{H}_{ij}(\boldsymbol{\mu}) = \sum_{n=1}^{N} \alpha_n(\boldsymbol{\mu})\, D_{ij}\alpha_n(\boldsymbol{\mu}).$$

The term $D_{ij}\alpha_n(\boldsymbol{\mu})$ describes the anisotropy of the n-th mode while $\alpha_n(\boldsymbol{\mu})$ indicates the its weight in the approximation.

2 **On-line construction of the metric.** Defining a mesh consisting of P-dimensional simplices with vertices being the set $\Xi_{\mathtt{trial}}$ allows to interpolate/represent quantities only defined on $\Xi_{\mathtt{trial}}$ by means of $P1$-finite element functions on the mesh for any parameter value $\boldsymbol{\mu} \in \mathbb{P}$, since the convex hull of $\Xi_{\mathtt{trial}}$ coincides with the parameter space $\mathbb{P}$. We can therefore easily access the interpolated Hessian $\overline{H}(\boldsymbol{\mu})$, for any $\boldsymbol{\mu} \in \mathbb{P}$, by means of its interpolated component functions $\overline{H}_{ij}(\boldsymbol{\mu})$.

Then, the construction of $d(\boldsymbol{\mu}_1, \boldsymbol{\mu}_2)$ is based on the Hessian at the points $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2$. Indeed, we will consider an average of the two local Hessian $\overline{H}(\boldsymbol{\mu}_i)$, $i = 1, 2$, to construct

$$H_{\boldsymbol{\mu}_1\boldsymbol{\mu}_2} = \frac{\overline{H}(\boldsymbol{\mu}_1)}{\|\overline{H}(\boldsymbol{\mu}_1)\|_F} + \frac{\overline{H}(\boldsymbol{\mu}_2)}{\|\overline{H}(\boldsymbol{\mu}_2)\|_F},$$

where $\|\cdot\|_F$ denotes the Frobenius norm.

The choice of this linear combination between $\overline{H}(\boldsymbol{\mu}_1)$ and $\overline{H}(\boldsymbol{\mu}_2)$ is motivated by the following argument. The Hessian is derived from a finite difference stencil and the order of the coefficients is not known and might vary in parameter space. In order to take equal quantities, we propose therefore to normalize the Hessian. This does not result in a loss of information since all we want to extract from the Hessian is the local anisotropy, but averaged properly.

One can then perform an eigenvalue decomposition

$$H_{\boldsymbol{\mu}_1\boldsymbol{\mu}_2} = V\,\Lambda\,V^T$$

where V is an orthogonal and $(\Lambda)_{ii} = \lambda_i$ a diagonal matrix consisting of the eigenvalues λ_i. Consider the diagonal matrix

$$|\Lambda|_{ii} = \frac{|\lambda_i|}{\sqrt{\lambda_1^2 + \ldots + \lambda_P^2}}, \qquad i = 1, \ldots, P,$$

and the associated symmetric positive definite matrix $M_{\boldsymbol{\mu}_1\boldsymbol{\mu}_2} = V\,|\Lambda|\,V^T$ to define the semi-distance between $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2$ by

$$d(\boldsymbol{\mu}_1, \boldsymbol{\mu}_2) = \sqrt{(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^T M_{\boldsymbol{\mu}_1\boldsymbol{\mu}_2}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)}.$$

5. Varying train sets based on Hessian. The goal of this section is to present an approach that further allows to reduce the offline computational cost. Here, we focus on the trial set $\Xi_{\texttt{trial}}$. It is aimed to keep its cardinality as small as possible, but large enough to capture the local geometry of the parametrized system. Since not only the cardinality of $\Xi_{\texttt{trial}}$ but also the locations of its points matters we propose to use the above constructed metric to design problem adapted training sets of appropriate size which will increase with increasing number of basis functions selected.

Firstly, we construct at each iteration of the presented algorithm a new trial set $\Xi_{\texttt{trial}}$. The cardinality of the $\Xi_{\texttt{trial}}$ being variable and an increasing function of the inverse of the actual error (-estimation) $\texttt{err}$. Let Q_m be the minimal cardinality of $\Xi_{\texttt{trial}}$ at the beginning of the algorithm and Q_M the maximal target cardinality at the end of the algorithm once the tolerance criteria $\texttt{err} < \texttt{tol}$ is reached, then we define

$$Q(\texttt{err}) = \left\lceil \frac{Q_M - Q_m}{\log(\texttt{tol})} \log(\texttt{err}) + Q_m \right\rceil$$

to be the number of points at the next iteration. Having the number of elements in $\Xi_{\texttt{trial}}$ for the next iteration specified it remains to construct an updated pointset $\Xi_{\texttt{trial}}$.

The trial set $\Xi_{\texttt{trial}}$ is constructed such that edges of the corresponding unique Delaunay triangulation are uniform in the slightly modified metric

$$\tilde{d}(\boldsymbol{\mu}_1, \boldsymbol{\mu}_2) = \frac{d(\boldsymbol{\mu}_1, \boldsymbol{\mu}_2)}{r(\boldsymbol{\mu}_{12})},$$

where $\boldsymbol{\mu}_{12} = \frac{1}{2}(\boldsymbol{\mu}_1 + \boldsymbol{\mu}_2)$. Taking the metric $\tilde{d}(\cdot,\cdot)$ instead of $d(\cdot,\cdot)$ is motivated by the fact that a ball defined by (3.1) corresponds (approximately) to a unit ball in $\tilde{d}(\cdot,\cdot)$. This modification of the metric allows to compare unit balls at a global level.

REMARK 5.1. *Enriching the trial set during the sampling process is not a new idea. An adaptively enriching greedy sampling strategy was proposed in [9]. We however want to mention the featurse that distinguishes our approach from the previously proposed version on a conceptual level. First our approach not only detects where the manifold of the solutions (as a function of the parameters) is complex but also the directions that are important to sample and those that are not. Second in our approach the size of the discrete system to be solved on-line is determined by the user leading to a controlled simulation cost.*

6. On-line implementation. It is well known that the stable implementation of the reduced basis method requires the orthonormalization of the snapshots $v(\boldsymbol{\mu})$, $\boldsymbol{\mu} \in \mathbb{S}$ [12, 14]. In the standard approach, this orthonormalization is prepared off-line and the vectorial space $\mathbb{W}$ spanned by the series $\{v(\tilde{\boldsymbol{\mu}}) \,|\, \tilde{\boldsymbol{\mu}} \in \mathbb{S}\}$ is not affected by this change of basis. In our alternative method, there are a large number of approximation spaces, almost one for each $\boldsymbol{\mu}$: $\mathbb{W}_{\boldsymbol{\mu}} = \text{span}\{v(\tilde{\boldsymbol{\mu}}) \,|\, \tilde{\boldsymbol{\mu}} \in \mathbb{S}_{\boldsymbol{\mu}}\}$. We cannot compute all various orthonormalized basis sets for all possible $\mathbb{W}_{\boldsymbol{\mu}}$ off-line. This would be too costly and using way too much memory.

Nevertheless, we can still *prepare* the orthonormalization process so that, the on-line orthonormalization is feasible. For any $\boldsymbol{\mu}$ and $\boldsymbol{\mu}'$ in $\mathbb{S}$, let us compute off-line the scalar products

$$\ell_{\boldsymbol{\mu},\boldsymbol{\mu}'} = \langle v(\boldsymbol{\mu}), v(\boldsymbol{\mu}') \rangle \tag{6.1}$$

where $\langle \cdot, \cdot \rangle$ represents an appropriate scalar product : say either L^2 or H^1 type. Note that we can even compute only those scalar products corresponding to pairs of parameters $(\boldsymbol{\mu}, \boldsymbol{\mu}')$ that are close, indeed if $\boldsymbol{\mu}$ and $\boldsymbol{\mu}'$ are distant in $\mathbb{P}$, they will never be in the same $\mathbb{S}_{\tilde{\boldsymbol{\mu}}}$.

In the on-line stage, once the approximation in $\mathbb{W}_{\boldsymbol{\mu}}$ will be required, we can easily orthonormalize the basis set $\{v(\tilde{\boldsymbol{\mu}}) \,|\, \tilde{\boldsymbol{\mu}} \in \mathbb{S}_{\boldsymbol{\mu}}\}$ first by declaring an order in the parameters

$$\{\tilde{\boldsymbol{\mu}} \in \mathbb{S}_{\boldsymbol{\mu}}\} = \{\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, ..., \boldsymbol{\mu}_N\} \tag{6.2}$$

then perform a classical Gramm-Schmidt orthonormalization process

$$\begin{aligned} \zeta_1 &= \beta_{1,1} v(\boldsymbol{\mu}_1), \\ \zeta_2 &= \beta_{2,2} v(\boldsymbol{\mu}_2) + \beta_{2,1} \zeta_1, \\ &\vdots \\ \zeta_N &= \beta_{N,N} v(\boldsymbol{\mu}_N) + \sum_{i=1}^{N-1} \beta_{N,i} \zeta_i, \end{aligned} \tag{6.3}$$

where the coefficients $\beta_{n,i}$ are chosen so that ζ_n is orthogonal to

$$\mathrm{span}\{v(\boldsymbol{\mu}_i) \,|\, i = 1, .., n-1\} = \mathrm{span}\{\zeta_i \,|\, i = 1, .., n-1\}$$

and the coefficients $\beta_{n,n}$ are chosen so that ζ_n is norm 1. It is well known that these coefficients exist, are unique and that, in order to compute them, we have to know each scalar product $\langle v(\boldsymbol{\mu}_n), \zeta_i \rangle$, $i = 1, .., n-1$. It should be noticed that computing those coefficients $\langle v(\boldsymbol{\mu}_n), \zeta_i \rangle$ on-line depends on the length of the basis vectors representing the basis functions $v(\boldsymbol{\mu}_n)$, which should be avoided during the on-line process. Instead, these scalar products can be easily computed from the precomputed coefficients $\ell_{\boldsymbol{\mu}_i, \boldsymbol{\mu}_j}$.

First, notice that the basis functions $\{\zeta_n\}_{n=1}^N$ can alternatively be expressed by the following linear transformation

$$\begin{aligned} \zeta_1 &= \gamma_{1,1} v(\boldsymbol{\mu}_1), \\ \zeta_2 &= \gamma_{2,2} v(\boldsymbol{\mu}_2) + \gamma_{2,1} v(\boldsymbol{\mu}_1), \\ &\vdots \\ \zeta_N &= \gamma_{N,N} v(\boldsymbol{\mu}_N) + \sum_{i=1}^{N-1} \gamma_{N,i} v(\boldsymbol{\mu}_i), \end{aligned} \tag{6.4}$$

from the set of basis functions $\{v(\boldsymbol{\mu}_n)\}_{n=1}^N$. It is our intention to construct the coefficients $\gamma_{n,i}$ representing the change of basis functions from $\{v(\boldsymbol{\mu}_n)\}_{n=1}^N$ to $\{\zeta_n\}_{n=1}^N$ as defined by (6.4) in a stable way and without any operation depending on the length of the vectors representing $v(\boldsymbol{\mu}_n)$ to guarantee stability and good on-line performance.

REMARK 6.1. *Note that the coefficients $\gamma_{n,i}$ could be obtained based on* (6.4) *directly. Indeed, each ζ_n is orthogonal to span$\{v(\boldsymbol{\mu}_i) \,|\, i = 1, .., n-1\}$ allowing to compute the coefficients $\gamma_{n,i}$, $i = 1, \ldots, n-1$. This however involves solving a linear systems based on the mass matrix with coefficients $\ell_{\boldsymbol{\mu}_i, \boldsymbol{\mu}_j} = \langle v(\boldsymbol{\mu}_i), v(\boldsymbol{\mu}_j) \rangle$, $i, j = 1, .., n-1$, which can be heavily linearly dependent leading to an unstable scheme.*

The coefficients $\beta_{n,k}$ can be obtained by the following formula mimicking the Gram-Schmidt procedure (6.3) but using the precomputed coefficients $\ell_{n,k}$. For the

construction of the coefficients $\beta_{n,j}$, we proceed as follows (it helps to have in mind that $\tilde{\ell}_{j,i}$ stands for $\langle \zeta_j, v(\boldsymbol{\mu}_i)\rangle$)) : for all $n = 1, \ldots, N$ apply recursively
First set

$$\tilde{\beta}_{n,n} = 1, \qquad \tilde{\beta}_{n,k} = -\tilde{\ell}_{k,n}, \qquad k = 1, \ldots, n-1,$$

so that $v(\boldsymbol{\mu}_n) - \sum_{k=1}^{n-1} \tilde{\beta}_{n,k}\zeta_k$ is orthogonal to $\text{span}\{\zeta_i \,|\, i = 1, .., n-1\}$. Then

$$\begin{aligned}
\alpha &= \left(\ell_{n,n} - \sum_{k=1}^{n-1} (\tilde{\ell}_{k,n})^2\right)^{\frac{1}{2}}, \\
\beta_{n,k} &= \tilde{\beta}_{n,k}/\alpha, \qquad k = 1, \ldots, n-1, \\
\tilde{\ell}_{n,k} &= \beta_{n,n}\ell_{n,k} + \sum_{j=1}^{n-1} \beta_{n,j}\tilde{\ell}_{j,k}, \qquad k = 1, \ldots, N.
\end{aligned}$$

Note that the construction of α is not subject to instabilities since the sum involves only positive values.

In theory, the set of coefficients $\{\gamma_{n,i}\}_{n,i=1}^N$ can be obtained by $\mathcal{O}(N^3)$ operations from the coefficients $\beta_{n,j}$, through a triangular process. Indeed, the coefficients $\gamma_{n,i}$ can be obtained by the following recursive formula

$$\gamma_{n,i} = \begin{cases} \sum_{k=1}^{n-1} \beta_{n,k}\gamma_{k,i} & \text{if} \quad i < n, \\ \beta_{n,n} & \text{if} \quad i = n, \\ 0 & \text{if} \quad i > n, \end{cases} \qquad \text{for all } n = 1, \ldots, N.$$

In practice however, the set of coefficients $\{\gamma_{n,i}\}_{n,i=1}^N$ is even not required, as explained in the following.

As mentioned above the set of basis functions $\{\zeta_n\}_{n=1}^N$ leads to well-conditioned matrices, the precomputed quantities are nevertheless expressed in the basis $\{v(\boldsymbol{\mu}_n)\}_{n=1}^N$. As an illustration assume that the reduced basis problem to be solved is about the approximation of the solution to a parametrized PDE, written in a variational form as : for a given parameter value $\boldsymbol{\mu}$, find u such that

$$a(u, v; \boldsymbol{\mu}) = f(v; \boldsymbol{\mu})$$

where it is assumed, for the sake of simplicity that a and f are affine decomposable

$$a(w, v; \boldsymbol{\mu}) = \sum_{p=1}^{P} g_p(\boldsymbol{\mu}) a_p(w, v), \quad f(v; \boldsymbol{\mu}) \sum_{q=1}^{Q} h_q(\boldsymbol{\mu}) f_q(v).$$

The ultimate goal is to construct the well-conditioned stiffness matrix $a(\zeta_i, \zeta_j; \boldsymbol{\mu})$ derived from the off-line pre-computation of the series $\alpha_{i,j}^p = a_p(v(\boldsymbol{\mu}_i), v(\boldsymbol{\mu}_j))$, for $1 \le i, j \le N$. To do so, we first compute the series $\tilde{\alpha}_{i,j}^p = a_p(v(\boldsymbol{\mu}_i), \zeta_j)$, for $1 \le i \le N$, recursively over j, $1 \le j \le N$:

$$\tilde{\alpha}_{i,j}^p = a_p(v(\boldsymbol{\mu}_i), \zeta_j) = \beta_{j,j}\alpha_{i,j}^p + \sum_{k=1}^{j-1} \beta_{j,k}\tilde{\alpha}_{i,k}^p.$$

Then, compute the series $a_p(\zeta_i, \zeta_j)$, for $1 \le i \le N$, recursively over j, $1 \le j \le N$:

$$a_p(\zeta_i, \zeta_j) = \beta_{j,j}\tilde{\alpha}_{i,j}^p + \sum_{k=1}^{j-1} \beta_{j,k} a_p(\zeta_k, \zeta_j), \tag{6.5}$$

before finally assembling the entries of the stiffness matrix

$$a(\zeta_i, \zeta_j; \boldsymbol{\mu}) = \sum_{p=1}^{P} g_p(\boldsymbol{\mu}) a_p(\zeta_i, \zeta_j), \tag{6.6}$$

that involves $\mathcal{O}(PN^2)$ operations.

Similarly, the evaluation of the each component of the vector $f(\zeta_i, \boldsymbol{\mu})$, $1 \leq i \leq N$ can be performed in $\mathcal{O}(PN)$ operations, and the inversion of the associated matricial problem is done in $\mathcal{O}(N^2)$ further operations.

7. Numerical results. Before we start with the numerical tests, an abstract description of the entire proposed local greedy algorithm is given in the following box:

```
LocallyAdaptiveGreedy
    ========== Stage 1 ==========
    1. Perform a classical greedy algorithm to select N + 1 basis functions.
    ========== Stage 2 ==========
    2. Compute the error estimate η(μ, W_μ) at each point μ ∈ Ξ_trial.
    3. Compute the metric function d.
    4. Enrich the set of basis functions.
    5. Create a new trial set Ξ_trial.
    6. Go to 2. until tolerance tol is achieved.
```

Algorithm: Locally adaptive greedy algorithm.

In a first step, Section 7.1, we only test the feature of local anisotropic approximation balls without adapting the train set using the metric. In a second step, Section 7.2, we test the entire algorithm.

As a reduced basis model $\mathbb{W}$, we use a parametrized family of functions combined with the L^2-projection P. The error is computed in an exact manner and no *a posteriori* estimation is used. For all numerical tests, Ω and $\mathbb{P}$ are discretized by a regular lattice of 75×75 points and the target tolerance is set to 10^{-3} if not otherwise mentioned.

7.1. Numerical results with fixed train set.

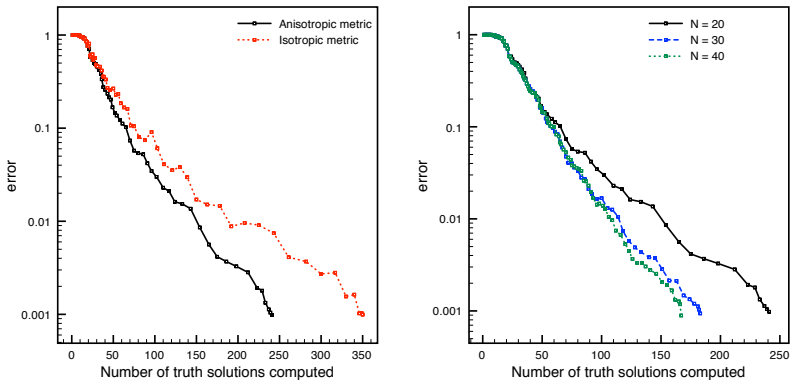
7.1.1. Test 1. We start with presenting a numerical example to illustrate the benefit of the local anisotropic approximation spaces. Consider the function

$$f_1(\boldsymbol{x}; \boldsymbol{\mu}) = \exp\left[\frac{-(x_1 - 0.1(\mu_1 - \mu_2))^2}{0.01} - \frac{(x_2 - (\mu_1 + \mu_2))^2}{0.01}\right],$$
$$\boldsymbol{x} \in \Omega = (-1, 1)^2, \boldsymbol{\mu} \in \mathbb{P} = [-0.5, 0.5]^2$$

that exhibits a constant anisotropy of parameters over the whole parameter space. The dependency in the $(\mu_1 + \mu_2)$-direction is ten times stronger than in the $(\mu_1 - \mu_2)$-direction. Figure 7.1 (left) illustrates the evolution of the number of basis functions that are necessary to be computed versus the achieved accuracy in the L^∞-norm for $N = 20$ for the anisotropic approach and the results of the same algorithm but using isotropic local approximation spaces as comparison. One can observe that using the anisotropic approach is beneficial in terms of the number of truth solutions to be computed at the off-line stage. Figure 7.1 (right) presents the same quantity but for varying $N = 20, 30, 40$. Not surprisingly the number of needed basis functions

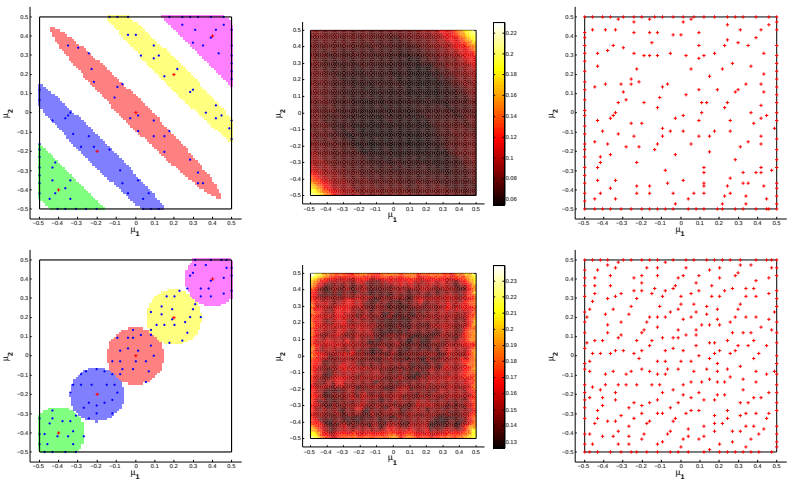
decreases while increasing N and more and more the exponential convergence establishes as for the classical Greedy-algorithm is expected. We can also observe that for a lower N more basis functions can be included per iteration.

Figure 7.2 illustrates the local approximation spaces, the radius as a function of the parameter and the sample set $\mathbb{S}$ for the anisotropic and the isotropic approach.



Comparison of different approaches. Comparison under variation of N.

Fig. 7.1: Test 1: Accuracy with respect to the number of truth solutions to be computed in comparison with the isotropic approach for $N = 20$ (left) and the for different values of $N = 20, 30, 40$ (right).



Local approximation spaces. Radius. Sample points.

Fig. 7.2: Test 1: Local approximation spaces for selected parameter values (left), radius as a function of the parameters (middle) and sample points (right) for anisotropic approach (top) and the isotropic version (bottom, as comparison) with $N = 20$.

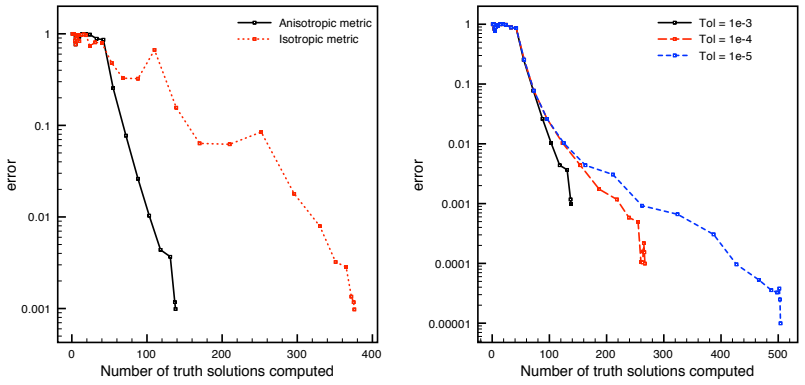
7.1.2. Test 2. The previous example is in some sense idealized since the anisotropy is not varying throughout the parameter domain. In this regard, the next example

$$f_2(\boldsymbol{x};\boldsymbol{\mu}) = \exp\left[-\frac{(x_1-(\mu_1^2+\mu_2^2))^2}{0.01} - \frac{(x_2-(\mu_1^2+\mu_2^2))^2}{0.01}\right], \quad \boldsymbol{x}\in\Omega=(-1,1)^2, \boldsymbol{\mu}\in\mathbb{P}=[-0.5,0.5]^2$$

is more interesting. It presents a family of parametrized functions where the functions (as functions of $\boldsymbol{x}$) are constant along concentric circles around the origin in parameter space. As an example, all four corners of the parameter space define the same function.

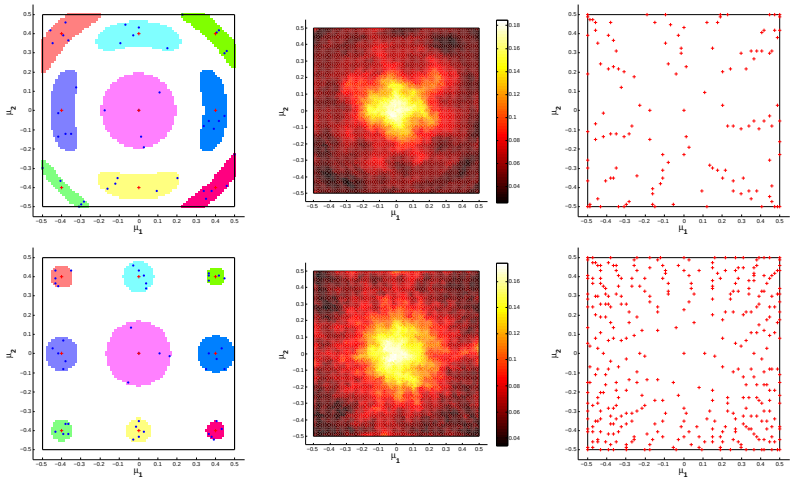
We compare again the number of required truth solutions for the anisotropic approach compared with the presented algorithm but using isotropic approximation spaces, which is presented in Figure 7.3 (left) for $N = 5$. We observe that 60% of the number of truth solutions to be computed can be saved. The corresponding local approximation spaces, radii and sample points $\mathbb{S}$ are illustrated in Figure 7.4. Figure 7.3 (right) also presents the number of required truth solutions for $N = 5$ but for different tolerance levels. We observe that starting from a critical value N_c of computed truth solutions, that depends on the tolerance, the curves show different behavior. During the iterations before N_c the complete parameter region has to be scanned because the tolerance is achieved nowhere in the parameter space. During the iterations after N_c however the tolerance is achieved in an increasing region of the parameter space and this part needs no longer to be not scanned. The enrichment of new truth solutions is then limited in space and the final phase of the greedy algorithm converges faster.

Additionally we present in Figure 7.5 the local approximation spaces for the different tolerance levels. One can clearly see how the approximation spaces take into account the local geometry that also allow non-convex approximation spaces.



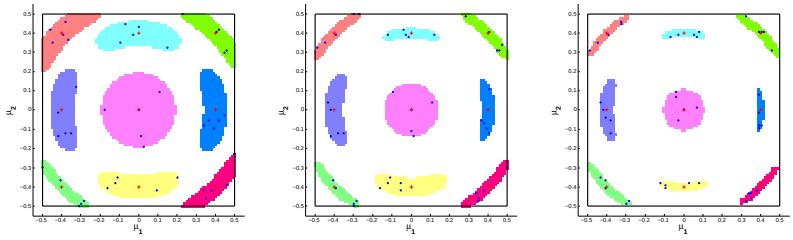
Anisotropy vs. Isotropy Different tolerances

Fig. 7.3: Test 2: Accuracy with respect to the number of truth solutions to be computed in comparison with the isotropic approach (left) and the for different end tolerances (right). In both cases $N = 5$.



Local approximation spaces. Radius. Sample points.

Fig. 7.4: Test 2: Local approximation spaces for selected parameter values (left), radius as a function of the parameters (middle) and sample points (right) for the presented approach (top) in comparison wit the isotropic version (bottom) for $N = 5$.



Tol=10^{-3} Tol=10^{-4} Tol=10^{-5}

Fig. 7.5: Test 2: Local approximation spaces for different tolerance levels.

7.1.3. Test 3. The next example

$$f_3(\boldsymbol{x};\boldsymbol{\mu}) = \exp\left[-\frac{(x_1-(\mu_1+3\mu_2))^2}{0.1+5|\mu_1+3\mu_2|} - \frac{(x_2-(3\mu_1-\mu_2))^2}{0.1+5|3\mu_1-\mu_2|}\right],$$
$$\boldsymbol{x}\in\Omega=(-1,1)^2, \boldsymbol{\mu}\in\mathbb{P}=[-0.5,0.5]^2$$

is interesting in the sense that it presents an almost singularity in parameter space at the origin. The performance of the anisotropic approach, compared to the isotropic version, is presented in Figure 7.6. Again, the anisotropic approaches outperform the isotropic one and about a 56% of computations of truth solutions can be saved.

The local approximation spaces, the radius and the sample points with $N = 10$ is presented in Figure 7.7. We observe *a posteriori* that the trial space $\Xi_{\texttt{trial}}$ was not sampled fine enough. Indeed in the region around the origin (and the cross for the isotropic version), every trial point is included in the set of sample points. This

also explains the sudden drop of the convergence in Figure 7.6, in particular for the isotropic version. This is a known problematic in greedy methods and our approach with adapted/moving trial sets presents a solution to this problem. The corresponding numerical results are presented in the next section. Further we recognize that the scheme detects the cross where the behavior of the function is most singular as can be seen by the chosen sample points in Figure 7.8 (right). Additionally, we present the sample points that were chosen for different functions of the type

$$f_{3,\xi_1,\xi_2}(\boldsymbol{x};\boldsymbol{\mu}) = \exp\left[-\frac{(x_1-\xi_1(\boldsymbol{\mu}))^2}{0.1+5|\xi_1|} - \frac{(x_2-\xi_2(\boldsymbol{\mu}))^2}{0.1+5|\xi_2|}\right]$$
$$\boldsymbol{x}\in\Omega=(-1,1)^2, \boldsymbol{\mu}\in\mathbb{P}=[-0.5,0.5]^2$$

where ξ_1 and ξ_2 are functions of $\boldsymbol{\mu}=(\mu_1,\mu_2)$. We can observe the adaptive nature of the algorithm.

Again, we plot the local approximation spaces for different values of the tolerance levels in Figure 7.9. We observe that the originally non-connected approximation spaces are becoming connected with increased tolerance.

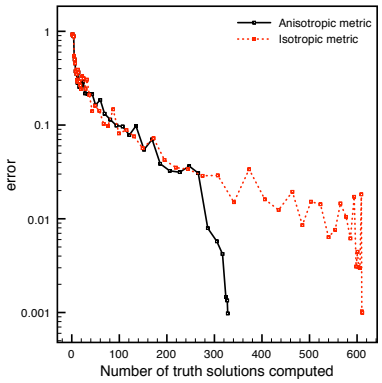


Fig. 7.6: Test 3: Accuracy with respect to the number of truth solutions to be computed in comparison with the isotropic approach for $N = 10$.

7.2. Numerical results with adapted train sets. In order to compare the accuracy of this version we introduce a fixed test set of sample points $\Xi_{\texttt{test}}$. In the following examples it consists of a lattice of 75×75 points. It corresponds to the fixed trial set of the tests of the previous section. We use FreeFem [10] in order to adapt the sample points by constructing the desired uniform mesh in the new metric.

It is not expected that the new approach with an adapting trial set uses fewer computed solutions than the approach using a fixed trial set since a non-adapted trial set may leave large errors in parameters that are outside the fixed trial set. On the contrary the adapted set tracks these forgotten parameter values. The benefit will be on the computational efficiency using the framework of a small trial set in the beginning that is gradually increasing. The points of the trials set are however chosen wisely using the information available from the local geometry of the system.

Figure 7.10 illustrates the total number of basis functions that are necessary to be computed versus the achieved accuracy of the proposed algorithm with adapting trial sets for the test cases 1, 2 and 3. In each plot, we present the error on the moving

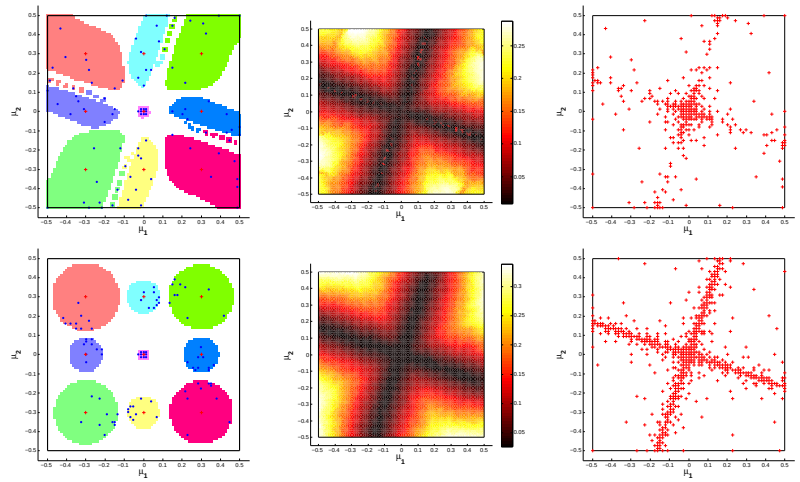
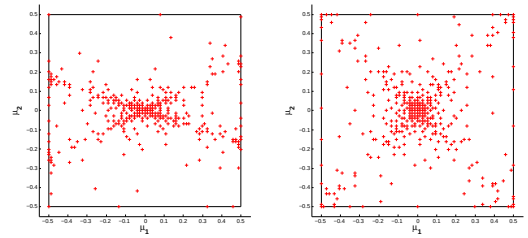


Fig. 7.7: Test 3: Local approximation spaces for selected parameter values (left), radius as a function of the parameters (middle) and sample points (right) for the presented approach (top) in comparison wit the isotropic version (bottom) for $N = 10$.



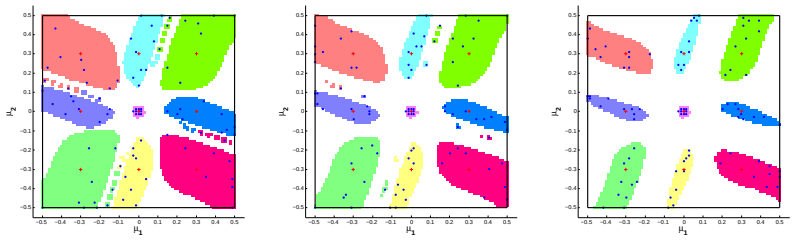
$\xi_1 = \mu_1 + 3\mu_2, \xi_2 = \frac{1}{2}\mu_1 - \mu_2$ $\xi_1 = \mu_1 + \mu_2, \xi_2 = \mu_1 - \mu_2$

Fig. 7.8: Sample set patterns for different type of parametrized functions with singular behavior on two crossing line with different slopes.

trial set, the error on the fixed testing set $\Xi_{\texttt{test}}$ and the accuracy of the version with a fixed trial set (which equals $\Xi_{\texttt{test}}$) as comparison.

7.2.1. Test 1. From Figure 7.10 (left), one can observe that the error on the trial set and on the test set of the approach with an adapted trial set are similar. This illustrates that the accuracy is not only guaranteed on the adaptive trial set but also satisfied on the test grid. Both errors are also similar to the error of the algorithm using a fixed trial set as presented in Section 7.1.

The adaptive version is as good as the version on a fixed trial set, but uses less evaluation of the error estimate as is shown in the following table:



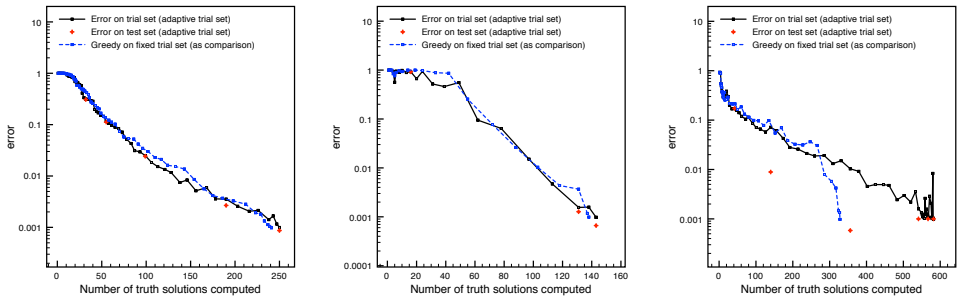
Tol=10^{-3} Tol=10^{-4} Tol=10^{-5}

Fig. 7.9: Test 3: Local approximation spaces for different tolerance levels.

Number of trial point evaluations for the greedy using adapted trial sets:	133'866
Number of trial point evaluations for the greedy using fixed trial sets:	382'500

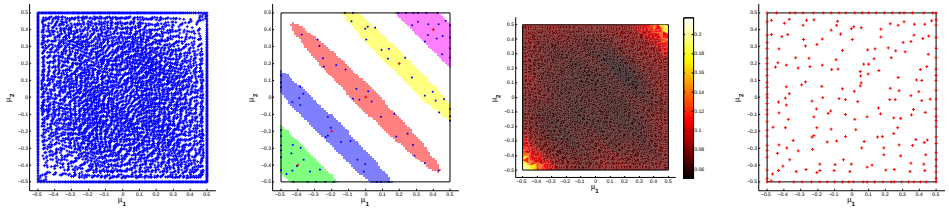
We observe a gain of 65% fewer error evaluations.

Figure 7.11 presents the trial set, local approximation spaces, radii and sample points for this calculation.



Test 1: $N = 20$ Test 2: $N = 5$ Test 3: $N = 10$

Fig. 7.10: Accuracy with respect to the number of truth solutions that need to be computed for all three test case.



Trial points. Local approximation spaces. Radius. Sample points.

Fig. 7.11: Test 1: Trial points, local approximation spaces for some specific parameter values, radii and sample points for greedy algorithm with adapted trial set and $N = 20$.

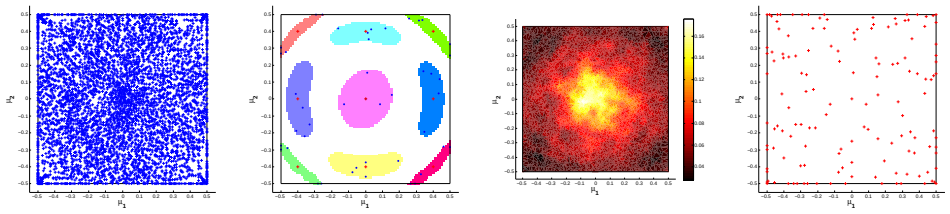
7.2.2. Test 2. Next, we consider the second example of the previous section. Figure 7.10 (middle) compares this version (with adapted trial sets) with the version of the previous section using a fixed trial set. One can observe again that the performance is similar than the version using a fixed trial set, and that the accuracy is also satisfied on the fixed test set $\Xi_{\mathtt{test}}$.

Comparing again with the version using a fixed trial set, the version with an adaptive trial set requires less evaluation of the error estimate as is shown in the following table:

Number of trial point evaluations during greedy using adapted trial sets:	37'444
Number of trial point evaluations during greedy using fixed trial sets:	123'750

We observe a gain of 70% fewer error evaluations.

Figure 7.12 presents the trial set, local approximation spaces, radii and sample points of this calculation.



Trial points. Local approximation spaces. Radius. Sample points.

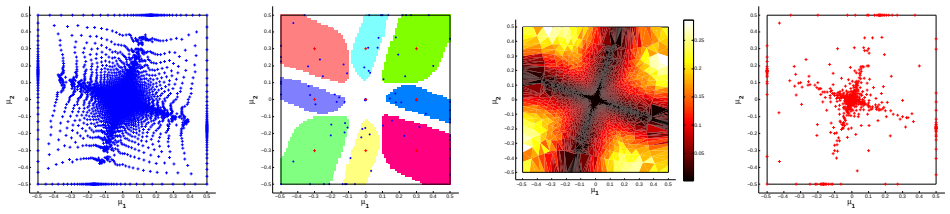
Fig. 7.12: Test 2: Trial points, local approximation spaces for some specific parameter values, radii and sample points for greedy algorithm with adapted trial set and $N = 5$.

7.2.3. Test 3. Finally, we apply the new algorithm with adapted trial set to the third example. As already mentioned, the solution of this example has a singularity and the version with a fixed train set did not resolve the singularity since the train set was not sampled fine enough. We expect the version with the problem-adapted train set to completely resolve the singularity, and thus in contrast using more sample points than the version with a fixed trial set. Figure 7.10 (right) presents the evolution of the number of sample points versus the accuracy of the algorithm. One can observe that this approach (with moving trial set) uses indeed more sample points than the version on a fixed train set as explained above. Finally, Figure 7.13 presents the trial set, local approximation spaces, radii and sample points. We observe that the trial points are in accordance with the sample points, and that a more dense sampling is indeed required around the origin. We observe that the local approximation spaces are now connected also for the tolerance of 10^{-3}.

In order to illustrate the benefit of using adaptive trial sets also in this case we consider a test sample of 75×75 uniformly distributed points in the region $[-0.05, 0.05]^2$ around the origin. Figure 7.14 illustrates the error distribution using the online procedure generated using a fixed (left) and an adaptive (right) trial set. The maximum error is 0.043 resp. 0.00146. While the error tolerance is almost satisfied in the latter case, it is clearly not the case for the former approach. Further, the number of error evaluations is given in the following table:

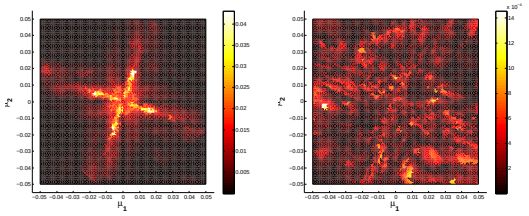
Number of trial point evaluations during greedy using adapted trial sets:	215'888
Number of trial point evaluations during greedy using fixed trial sets:	241'875

Thus, the adaptive version uses still less error estimator evaluations and is more accurate in the region around the origin. Also, the error is more equally distributed.



Trial points. Local approximation spaces. Radius. Sample points.

Fig. 7.13: Test 3: Trial points, local approximation spaces for some specific parameter values, radii and sample points for greedy algorithm with adapted trial set and $N = 10$.



Fixed trial set. Adaptive trial set.

Fig. 7.14: Test 3: Error distribution on a region around the origin of the greedy version using a fixed (left) and an adaptive (right) trial set.

8. Conclusions. We presented a generalization of the classical greedy algorithm that is used in the framework of reduced basis methods and the empirical interpolation method. The presented algorithm introduces local approximation spaces (in the parameter space) that also account for local anisotropic behavior instead of a global approach. The key idea was to consider the N closest basis functions (for a fixed N) where the semi-distance is measured in an empirically built metric which can be constructed on the fly and which is problem-dependent. Numerical tests illustrate the benefit of this approach.

Acknowledgement. The authors want to thank the help and discussions with Frederic Hecht about the adaptation of FreeFem++ to their purpose. This work was also initiated out of the discussions during the WORKSHOP JLL-SMP: "REDUCED BASIS METHODS IN HIGH DIMENSIONS" held in June 2011, organized with the support of the Fondation de Sciences Mathématiques de Paris.

REFERENCES

[1] M. Barrault, Y. Maday, N. C. Nguyen, and A. T. Patera. An 'empirical interpolation' method: application to efficient reduced-basis discretization of partial differential equations. *C. R. Math. Acad. Sci. Paris*, 339(9):667–672, 2004.

[2] P. Binev, A. Cohen, W. Dahmen, R. DeVore, G. Petrova, and P. Wojtaszczyk. Convergence rates for greedy algorithms in reduced basis methods. *SIAM J. Math. Anal.*, 43(3):1457–1472, 2011.

[3] A. Buffa, Y. Maday, A. T. Patera, C. Prud'homme, and G. Turinici. A priori convergence of the greedy algorithm for the parametrized reduced basis method. *Esaim-Math Model Num*, 46(3):595–603, Jan 2012.

[4] C. D. Cantwell, S. J. Sherwin, R. M. Kirby, and P. H. J. Kelly. From h to p efficiently: strategy selection for operator evaluation on hexahedral and tetrahedral elements. *Comput. & Fluids*, 43:23–28, 2011.

[5] I. Danaila and F. Hecht. A finite element method with mesh adaptivity for computing vortex states in fast-rotating bose-einstein condensates. *J. Comput. Phys.*, 229(19):6946–6960, 2010.

[6] J. L. Eftang, A. T. Patera, and E. M. Rønquist. An "*hp*" certified reduced basis method for parametrized elliptic partial differential equations. *SIAM J. Sci. Comput.*, 32(6):3170–3200, 2010.

[7] J. L. Eftang and B. Stamm. Parameter multi-domain 'hp' empirical interpolation. *Internat. J. Numer. Methods Engrg.*, 90(4):412–428, 2012.

[8] M. A. Grepl, Y. Maday, N. C. Nguyen, and A. T. Patera. Efficient reduced-basis treatment of nonaffine and nonlinear partial differential equations. *M2AN Math. Model. Numer. Anal.*, 41(3):575–605, 2007.

[9] B. Haasdonk, M. Dihlmann, and M. Ohlberger. A training set and multiple bases generation approach for parameterized model reduction based on adaptive grids in parameter space. *Mathematical and Computer Modelling of Dynamical Systems*, 17(4):423–442, 2011.

[10] F. Hecht, O. Pironneau, A. L. Hyaric, and K. Ohtsuka. *FreeFEM++ Manual*, 2008.

[11] L. Machiels, Y. Maday, I. B. Oliveira, A. T. Patera, and D. V. Rovas. Output bounds for reduced-basis approximations of symmetric positive definite eigenvalue problems. *C. R. Math. Acad. Sci. Paris*, 331(2):153–158, 2000.

[12] C. Prud'homme, D. V. Rovas, K. Veroy, Y. Maday, A. T. Patera, and G. Turinici. Reliable real-time solution of parametrized partial differential equations: Reduced-basis output bound methods. *Journal of Fluids Engineering*, 124(1):70–80, Mar 2002.

[13] S. Sen, K. Veroy, D. B. P. Huynh, S. Deparis, N. C. Nguyen, and A. T. Patera. "Natural norm" a posteriori error estimators for reduced basis approximations. *J. Comput. Phys.*, 217(1):37–62, 2006.

[14] K. Veroy and A. T. Patera. Certified real-time solution of the parametrized steady incompressible Navier-Stokes equations: Rigorous reduced-basis a posteriori error bounds. *Internat. J. Numer. Methods Engrg.*, 47(8-9):773–788, 2005.